Fehlerbehebung bei CVP VXML-Server-Speicherproblemen mit JConsole

Inhalt

Einleitung Voraussetzungen Anforderungen Verwendete Komponenten JConsole-Dienstprogramm

Einleitung

Dieses Dokument beschreibt die Verwendung des Java Console (jconsole)-Tools zur Fehlerbehebung bei Speicherlecks im Cisco Unified Customer Voice Portal (CVP).

Voraussetzungen

Anforderungen

Cisco empfiehlt, dass Sie über Kenntnisse in folgenden Bereichen verfügen:

- Cisco Unified Customer Voice Portal (CVP)
- Java Console-Dienstprogramm

Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf CVP Version 12.5.

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

JConsole-Dienstprogramm

Zur Fehlerbehebung bei Anrufservern, Speicherlecks im VXML-Server oder Leistungsproblemen ist es häufig erforderlich, die Java-Heap-Dump-Trace im Dienstprogramm jconsole zu aktivieren. Dies erfolgt in der Regel, nachdem Sie ein Ressourcenproblem mithilfe der Windows Event Viewer-, Task-Manager- und/oder Performance-Protokolltools auf einen bestimmten Dienst oder Dienste im Server beschränkt haben. Der dump-Heap für Dienstprogramme ist eine Trace-Datei mit niedriger Ebene und wird empfohlen, die Trace-Ebene für die CVP-Fehlerbehebung zu aktivieren.

Jconsole befindet sich standardmäßig im Pfad C:\Cisco\CVP\jre\bin des CVP-Servers. Die

Portdetails sind bereits standardmäßig in jmx_callserver.conf, jmx_vxml.conf, jmx_oamp.conf und jmx_wsm.conf Konfigurationsdateien unter C:\Cisco\CVP\conf für die jeweiligen CVP-Server konfiguriert.

- Anrufserver JMX-Port 2098
- VXML-Server JMX-Port 9696
- OAMP-Server JMX-Port 10001
- WSM JMX-Port 2099

🗾 jmx_callserver.conf - Notepad	🧾 jmx_vxml.conf - Notepad
File Edit Format View Help	File Edit Format View Help
<pre>com.sun.management.jmxremote.ssl.need.client.auth = false com.sun.management.jmxremote.authenticate = false com.sun.management.jmxremote.port = 2098 com.sun.management.jmxremote.ssl = false #com.sun.management.jmxremote.ssl.config.file= com.sun.management.jmxremote.rmi.port = 2097</pre>	<pre>com.sun.management.jmxremote.ssl.need.client.auth = false com.sun.management.jmxremote.authenticate = false com.sun.management.jmxremote.port = 9696 com.sun.management.jmxremote.ssl = false #com.sun.management.jmxremote.ssl.config.file= com.sun.management.jmxremote.rmi.port = 9697</pre>
jmx_oamp.conf - Notepad	////jmx_wsm.conf - Notepad
III jmx_oamp.conf - Notepad File Edit Format View Help	jmx_wsm.conf - Notepad File Edit Format View Help

Sie können das Jconsole-Dienstprogramm wie in den folgenden Schritten beschrieben ausführen:

Schritt 1: Navigieren Sie zu **%CVP_HOME%/CVP/jre/bin/jconsole.exe**, und doppelklicken Sie auf **jconsole.exe**.



Schritt 2: Stellen Sie eine Verbindung mit **localhost her**, und geben Sie die Portnummer für die CVP-Komponente an, mit der Sie eine Verbindung herstellen möchten. Beispiel: Für den CVP VXML-Server wird der JMX-Port 9696 verwendet. Lassen Sie das Feld **Benutzername** und **Kennwort** leer. Klicken Sie auf **Verbinden**.

🙆 Java Monitoring & Management Console	_	x
Connection Window Help		
🛃 JConsole: New Connection		
New Connection		
New Connection		
Remote Process:		
localhost:9696		
Usage: <hostname>:<port> OR service:jmx:<protocol>:<sap></sap></protocol></port></hostname>		
Username: Password:		
Connect Cancel		

Schritt 3: Klicken Sie auf Verbindung unsicher.



Schritt 4: Wählen Sie die Registerkarte MBeans aus.

🛃 Java Monitoring & Management Console - Administrator@localhost:9696	-		x
💁 Connection Window Help		_	Ξ×
Overview Memory Threads Classes VM Summary MBeans			-
Overview remove remove the set of			
< III >			

Schritt 5: Erweitern Sie **com.sun.management > HotSpotDiagnostic > Operations**, und klicken Sie auf **dumpHead**.

📓 Java Monitoring & Management Console - Administrator@localhost:9696 🗕 🗖 🗙					
🅌 Connection Window He	📓 Connection Window Help				
Overview Memory Threads O	Classes VM Summary MBea	ans	-		
Catalina Cisco Unified CVP VXML S Implementation Cisco Scources Commentation	Operation invocation void dumpHeap (MBeanOperationInfo	p0 String , p1 true)			
E. 🛞 DiagnosticCommand	Name	Value			
🖃 🧐 HotSpotDiagnostic	Operation:				
	Name	dumpHeap			
Operations	Description	dumpHeap			
dumpHeap	Impact	UNKNOWN			
getVMOption	Return i ype	Void			
setVMOption	Parameter-U:				
🗄 🔄 java.lang	Name				
🗄 🚽 java.nio	Description	pu inve lane String			
🗄 🔤 java.util.logging	Type Decemptor 1	java.lang.string			
i ⊕ org.apache.commons.po	Parameter-1:	-1			
	Description	p1			
	Description	p1			
	Descriptor				
	Name	Value			
	Operation:				
	openType	javax.management.openmbean.SimpleType(name=java.lang.Void)			
	originalType	void			
	Parameter-0:				
	openType	javax.management.openmbean.SimpleType(name=java.lang.String)			
	originalType	java.lang.String			
	Parameter-1:				
	openType	javax.management.openmbean.SimpleType(name=java.lang.Boolean)			
	originalType	boolean			
< III >					

Schritt 6: Geben Sie in **p0** den Dateinamen für den Dump mit der Erweiterung .hprof ein, z. B. **vxmlDump.hprof**. Lassen Sie **p1** als **true**.

📓 Java Monitoring & Management Console - Administrator@localhost:9696 🛛 📮 🗖 🗙					x
🖆 Connection Window Help					ъ×
Overview Memory Threads Classes VM Summary MBeans					-
Catalina Cisco Unified CVP VXML S JMImplementation Users com.cisco.ccbu	aration invocation d dumpHeap (p	00 vxmlDump.hprof , p1 true)			
com.sun.management	ne	Value			
⊕ · · · · · · · · · · · · · · · · ·	ration:				
HotSpotDiagnostic Nam	ie	dumpHeap			
Attributes Desc	cription	dumpHeap			
Impa Impa	act	UNKNOWN			
	urnType	void			
Para	ameter-0:				
H- iava.lang	e	p0			
java.nio Desc	cription	p0			
java.util.logging	e	java.lang.String			
org.apache.commons.po	ameter-1:				
Nam	ie	p1			
Desc	cription	p1			
Des	criptor				
Nar	ne	Value			
Oper	ration:				
oper	nType	javax.management.openmbean.SimpleType(name=java.lang.Void)			
origi	inalType	void			
Para	ameter-0:				
oper	nType	javax.management.openmbean.SimpleType(name=java.lang.String)			
origi	nalType	java.lang.String			
Para	imeter-1:				
oper	ni ype	javax.management.openmbean.simpleType(name=java.lang.boolean)			
ongi	папуре	bolean			

Anmerkung: Cisco empfiehlt, den Heap während der Geschäftszeiten nicht zu kippen, da dies dazu führen kann, dass der VXML-Server-Service für einen kurzen Moment während des Vorgangs nicht mehr verfügbar ist. Cisco empfiehlt, diese Aktivität außerhalb der Geschäftszeiten durchzuführen.

Schritt 7: Klicken Sie auf dumpHeap.

🛃 Java Monitoring & Management Console - Administrator@localhost:9696 🛛 🗖 🗙				
🛓 Connection Window Help				
Overview Memory Threads Classes VM Summary MBeans				
Catalina Cisco Unified CVP VXML S JMImplementation Users com.cisco.ccbu	Operation invocation void dumpHeap (MBeanOperationInfo	p0 vxmlDump.hprof , p1 true)		
com.sun.management	Name	Value		
HotSpotDiagnostic	Operation:			
Attributes	Name	dumpHeap		
	Description	dumpHeap		
dumpHeap	Impact	UNKNOWN		
getVMOption	ReturnType	void		
setVMOption	Parameter-0:			
java.lang	Name	p0		
🗄 🗤 📊 java.nio	Description	p0		
🖅 🚽 java.util.logging	Туре	java.lang.String		
主 🚽 org.apache.commons.po	Parameter-1:			
	Name	p1		
	Description	p1		
	Descriptor			
	Name	Value		
	Operation:			
	openType	javax.management.openmbean.SimpleType(name=java.lang.Void)		
	originalType	void		
	Parameter-0:			
	openType	javax.management.openmbean.SimpleType(name=java.lang.String)		
	originalType	java.lang.String		
	Parameter-1:	in an anna an an Anna Cinela Trans (anna - ina lana Barlana)		
	openiype	javax.management.openmbean.SimpleType(name=java.lang.boolean)		
	originari ype	boolean		
< III >				

Schritt 8: Die Meldung Method wurde erfolgreich aufgerufen. Klicken Sie auf OK.

🛃 Java Monitoring & Man	agement Console - Ad	ministrator@loca	lhost:9696
🕌 Connection Window H	lelp		
Overview Memory T	Info)	
Catalina Cisco Unified CV JMImplementati Users com.cisco.ccbu com.sun.manag	Method successf	ully invoked	nprof , p1 true)
E 🛞 DiagnosticCommand	Thome -	Value	
	Operation:		
	Name	dumpHeap	0
	Description	dumpHeap	5
	Impact	UNKNOWN	
act/MOction	ReturnType	void	
get/MOption	Parameter-0:		
setvMOption	Nin and	-0	

Schritt 9: Erfassen Sie die generierte Dump-Datei. Der Standardpfad für das VXML-Serverdump

ist C:\Cisco\CVP\VXMLServer\Tomcat\bin.

C:\Cisco\CVP\	VXMLServer\Tomcat\bin
Name	
vxmlDump.h	prof

Hinweis: Dieses Tool ist zur Fehlerbehebung bei Speicherlecks im VXML-Server vorgesehen. Sobald die Anwendung identifiziert wurde, die das Leck verursacht, muss dieses Tool aktiviert, die gewünschten Informationen müssen gesammelt und nach der Wiederherstellung des Problems muss es deaktiviert werden. Jconsole ist nicht als Monitortool konzipiert und darf nicht auf unbegrenzte Zeit aktiviert werden.